Real-Time Data Aggregation in Contention-Based Wireless Sensor Networks

JUN ZHANG and XIAOHUA JIA
City University of Hong Kong
and
GUOLIANG XING
Michigan State University

We investigate the problem of delay constrained maximal information collection for CSMA-based wireless sensor networks. We study how to allocate the maximal allowable transmission delay at each node, such that the amount of information collected at the sink is maximized and the total delay for the data aggregation is within the given bound. We formulate the problem by using dynamic programming and propose an optimal algorithm for the optimal assignment of transmission attempts. Based on the analysis of the optimal solution, we propose a distributed greedy algorithm. It is shown to have a similar performance as the optimal one.

Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Distributed networks*

General Terms: Algorithms, Design, Performance

Additional Key Words and Phrases: Sensor networks, data aggregation, real-time traffic, CSMA/CA, delay constrained transmission

ACM Reference Format:

Zhang, J., Jia, X., and Xing, G. 2010. Real-time data aggregation in contention-based wireless sensor networks. ACM Trans. Sensor Netw. 7, 1, Article 2 (August 2010), 25 pages. DOI = 10.1145/1806895.1806897 http://doi.acm.org/10.1145/1806895.1806897

1. INTRODUCTION

Data aggregation [Rajagopalan and Varshney 2006] is a widely accepted method for collecting data in sensor networks [Culler et al. 2004; Pottie and Kaiser 2000; Chen and Varshney 2004; Akyildiz et al. 2002]. The sensor nodes

This work was supported by a grant from the Research Grants Council of Hong Kong [Project No. CityU 7002359] and the US National Science Foundation under a CAREER award CNS-0954039.
Authors' addresses: J. Zhang, X. Jia, Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong, China; email: junzhang@cityu.edu.hk; jia@cs.cityu.edu.hk; G. Xing, Department of Computer Science and Engineering, Michigan State University, 3115 Engineering Building, East Lansing, MI 48824; email: glxing@msu.edu.

DOI 10.1145/1806895.1806897 http://doi.acm.org/10.1145/1806895.1806897

are organized into a data aggregation tree, where the root of the tree is the sink node. Data from sensor nodes is aggregated at intermediate nodes, combined with the local data, and forwarded up to the sink along the tree. By using data aggregation, the traffic over the network is substantially reduced, particularly at the nodes close to the sink. Extensive research has been done on data aggregation, such as reducing energy consumption [Intanagonwiwat et al. 2000; Tang and Xu 2008], prolonging the network life time [Al-Karaki et al. 2004; Kalpakis et al. 2003; Younis and Fahmy 2005; Xue et al. 2005; Pan and Lu 2007; Hua and Yum 2008], security [Perrig et al. 2002], minimizing latency [Intanagonwiwat et al. 2002; Krishanamachari et al. 2002; Solis and Obraczka 2004; Du et al. 2006; Yu et al. 2004; Huang et al. 2007], and QoS data aggregation [Zhu et al. 2006].

Many applications of sensor networks require real-time data collection. In such applications, a data packet that arrives at the sink outside of some required delay bound has no value. This is also true for the applications that collect data from sensor nodes periodically. The data from any sensor node should reach the sink within a period of data collection. TDMA-based technology [Chipara et al. 2006] is a natural choice for real-time sensor network applications, where all nodes are strictly synchronized and scheduled to transmit in a conflict-free fashion. However, most of the sensor nodes available on the market follow IEEE 802.15.4 standard and use CSMA/CA scheme for data transmission, such as EPIC Mote, an open mote platform for application-driven designing,[1] FlatMesh,[2] and Sunspot.[3] This motivates us to study the support of real-time sensor network applications in the CSMA-based model.

The key challenge of supporting real-time data transmission in CSMA-based model is the nondeterministic of delay for a successful transmission of a data packet. CSMA/CA is a contention-based technique. A node needs to sense whether the channel is idle before it can transmit a packet. If the channel is not idle at that time, the node needs to wait for a certain period of time before it senses the channel again. This scheme makes the delay time for a successful transmission nondeterministic. Apparently, for a real-time application, in order to meet end-to-end delay requirement we have to set a maximal allowable delay time for data transmission at each node. A data packet will be dropped if its delay exceeds the maximal allowable delay of this packet at a node. Due to the data aggregation, the amount of information contained in each data packet is different. The data packets that are aggregated from more sensor nodes contain more information and these packets are usually the ones that reach the high level of the aggregation tree and are close to the sink. The problem of our concern is, given the end-to-end delay bound of an application, how to allocate the maximal allowable transmission delay at each node of the aggregation tree, such that the amount of information collected at the sink is maximized and the total delay for the data aggregation is within the given end-to-end delay bound. The amount of information collected is defined

[1] `http://www.cs.berkeley.edu/prabal/projects/epic.`
[2] `http://www.senceive.com/Products/aspx.`
[3] `http://www.sunspotworld.com.`

as the number of sensor nodes whose data have been aggregated and reached the sink. Notice that our work is not comparable with those on minimizing latency, such as DB-MAC [Bacco et al. 2004], or improving throughput, such as funneling-MAC [Ahn et al. 2006], as our objective is to reach a good tradeoff between delay and throughput.

This is a difficult optimization issue. An optimal solution needs to consider several major factors: the amount of information in a packet, the delay time of its transmission, and its success probability. The issue is further complicated by the fact that the delay and success probability of a transmission depends on network interference, traffic load, and the aggregation tree structure. We first formulate the problem by using dynamic programming technique and propose an algorithm to find the optimal solution. Then, based on the analysis of the optimal solution and experimental results, we propose a distributed heuristic algorithm. Simulation results have shown that this distributed algorithm has a performance close to the optimal one. To the best knowledge of the authors, our article is the first in the literature to address the maximal information collection problem under the delay bound constraint for CSMA/CA protocols and we find the optimal solution.

The article is organized as follows. The problem is formulated in Section 2. The optimal algorithm to solve this problem is presented in Section 3. The distributed heuristic algorithm is presented in Section 4. The simulation results are shown in Section 5. The conclusion is given in Section 6.

2. PROBLEM FORMULATION

2.1 Query and Network Model

Suppose there are N sensor nodes $v_1, v_2, \ldots, v_N$ and one sink v_0 in the network. These nodes $V = (v_0, v_1, \ldots, v_N)$ form a data aggregation tree $T(v_0)$, where v_0 is the root of the tree. We consider real-time or periodical data aggregation applications. Suppose the end-to-end delay bound (or the period) for a data aggregation query is D_0. That is, the data from any sensor node shall reach the sink within a delay of D_0. The traffic generation rate of each node is λ, which depends on the frequency of the queries. For data aggregation, each nonleaf node in the aggregation tree must wait for data packets from all its children, before it aggregates them together with its own data and forwards the aggregated data to its parent toward the sink. We assume data packets sent by all nodes have the same size due to data aggregation. The amount of information contained in each packet may be different, depending on how much information is forwarded to the sink through the sender of the packet.

2.2 Basic Operation of Contention-Based MAC

We assume that the network employs a CSMA-based MAC layer such as IEEE 802.11 [IEEE Computer Society LAN MAN Standards Committee 1997] or IEEE 802.15.4 [IEEE Computer Society LAN MAN Standards Committee 2006] protocols. We first briefly describe the channel access operation of 802.11 (802.15.4 beaconless mode follows the similar scheme). A node performs a clear channel assessment (CCA) before transmitting a new packet. The protocol uses a contention window and the backoff mechanism to reduce the possibility of collision when two nodes sense idle and decide to transmit simultaneously. A fixed contention window or an exponential decrease in window size was suggested in Woo and Culler [2001] to provide proportional fairness for sensor networks. In particular, as reported by Römer et al. [2006], a fixed contention window is commonly adopted in sensor networks design, such as B-MAC [Polastre et al. 2004], due to its simplicity and good performance in most practical scenarios. Thus, we consider a variant of CSMA-based MAC with a fixed contention window w in this article. When a node has a packet ready to transmit, it chooses a random backoff counter, which is an integer in the range of $[0, (w-1)]$. The backoff counter decreases by 1 when the node senses an idle channel for a timeslot. When the backoff counter becomes 0, it will continue to monitor the channel to be idle for a certain duration of time, referred to as distributed interframe space (DIFS). It then transmits the packet. If either the channel becomes busy in the duration of DIFS or a collision occurs during the transmission, it goes back to reset the backoff counter and starts the whole process again. This is counted as one transmission attempt. The node keeps on trying to transmit the packet until either a transmission is successful or the number of attempts reaches a prespecified *maximal number of transmission attempts*. This maximal number of transmission attempts determines the expected worst-case delay that a packet is allowed at a node. The packet is either successfully transmitted before this delay or simply discarded. This maximal number of attempts also determines the success probability of transmitting a packet. The larger this number is, the more times the packet is given for retries and the higher probability of success it has. In this article, we will determine the maximal number of transmission attempts for each node in the data aggregation tree, such that the number of nodes whose data has been successfully transmitted to the sink is maximized and the total delay does not exceed the required bound.

2.3 Per-Hop Delay and Success Probability Analysis

In this section, we model the per-hop transmission delay and the successful probability in sensor networks. The transmission and interference model is based on the CSMA/CA protocol model. Suppose that a signal attenuation function $A(u, v)$ is given. Its value is the signal attenuation rate from sender u to receiver v. Let P_u denote node u's transmission power. According to the signal attenuation rate, the signal strength received at receiver v is $A(u, v)P_u$.

In order to let node v to decode the packet from node u, the following must be held:

$$A(u, v)P_u \geq \alpha,$$

where α is the decoding threshold of signal strength such that a receiver can decode the signal properly.

A node v, is said to be interfered by another node u, if the signal strength received by v from u exceeds a carrier sensing threshold β, that is,

$$A(u, v)P_u \geq \beta.$$

Any node v is not allowed to transmit simultaneously with its interfering node. According to our transmission and interference model, the transmission and interference range can either be spherical, or irregular, depending on the setting of the signal attenuation function $A(u, v)$. Hence our transmission and interference model is compatible with both the theoretical disk model in Alicherry et al. [2005], and the practical model based on measurement in Zhou et al. [2005]. The interference of node v in the network can be modeled by the contention model used for modeling wireless LANs. The fact that all nodes that interfere with v compete with each other for a transmission timeslot can be regarded as a set of nodes (mobile units) competing to access an assess point (AP) in a wireless LAN.

A Markov model was presented in Barowski et al. [2005] to analyze the unsaturated throughput and delay of wireless LANs of the IEEE802.11 protocol, based on the fundamental work of Bianchi [2000]. In this model, the link collision probability and the link delay in a wireless LAN are expressed as functions of the number of nodes and the traffic load generated by the nodes. Since this model considers binary increasing contention windows, the collision probability and the link delay also depend on the setting of minimal and maximal contention windows. We consider a special case of this model, such that the minimal window and the maximal window are both of a fixed size w, and it becomes a constant in our analysis. Given a wireless LAN of N nodes (mobile units) and traffic load from each node λ, we can calculate the link collision probability $P_c(N, \lambda)$, the time of a successful transmission $T_S(N, \lambda)$, and the time of a failed transmission attempt $T_F(N, \lambda)$ by the numerical method. For details of the calculation, we refer the reader to Barowski et al. [2005].

We employ this wireless LAN model to model the per-hop delay and success probability of data transmission in the sensor network. We adopt the notation in Barowski et al. [2005], and express the link successful probability and link delay for sensor networks by Equations (1), (2), and (3) below. We consider continuous queries in this article. It is possible that a parent node contends a channel with its child at same time, since it may transmit data aggregated in last round of queries (but not data in this round as it has not aggregated it from its children yet). Thus each node in the sensor network can be regarded as a sender with a constant data rate λ. In data aggregation, a node transmits data only to its parent in the aggregation tree. The link collision probability of v to its parent can be approximated as $P_c(X(v), \lambda)$, where $X(v)$ is the number of nodes that interfere with v. Let k denote the maximal number of transmission attempts for node v. That is, node v will keep on trying to transmit a packet to its parent, until either a transmission is successful or the number of failed transmissions reaches k. Let $p(v, k)$ denote the success probability for node v to transmit a packet to its parent with the maximal number of transmission attempts k. We have

$$p(v, k) = 1 - [P_c(X(v), \lambda)]^k. \tag{1}$$

Since the sink v_0 will not transmit data anymore, $p(v_0, k)$ is defined as 1.

Next we compute the per-hop delay. Again, we approximate the time for a successful transmission of a packet and the time for a failed attempt for node

v in a sensor network as that in the wireless LAN model. They are $T_S((X(v), \lambda)$ and $T_F((X(v), \lambda)$, respectively. When v transmits a packet to its parent i times to make it successfully received, it has experienced $i - 1$ transmission failures and one success. Thus, the total time for the i times of transmissions is

$$T_S(X(v), \lambda) + (i - 1)T_F(X(v), \lambda). \tag{2}$$

The expected link delay from v to its parent with the maximal number of transmission attempts k, denoted as $d(v, k)$, is

$$d(v, k) = \sum_{i=1}^{k} [P_c(X(v), \lambda)]^{i-1}(1 - P_c(X(v), \lambda))(T_S((X(v), \lambda) + (i - 1)T_F((X(v), \lambda)). \tag{3}$$

Similarly, since v_0 is the sink node, $d(v_0, k)$ is defined as zero.

With per-hop delay $d(v, k)$ and success probability $p(v, k)$ under the maximal number of transmission attempts k, we can formulate the problem of maximizing the number of nodes that have their data successfully received by the sink within the end-to-end delay bound in the sensor network system.

2.4 Problem Formulation

Given a data aggregation tree of N sensor nodes $v_1, v_2, \ldots, v_N$ and one sink v_0, we consider the problem of how to assign the maximal number of transmission attempts for each node, so that the amount of information collected at the sink is maximized and the maximal end-to-end delay is within the given bound D_0. To make sure that the maximal number of transmission attempts are not infinity, we set M as their upper limit.

First of all, we define the transmission attempts assignment of the sensor network. Let $T(v)$ denote the subtree rooted at node v. We denote

$$H_{T(v)} = \{(u, k_u) : u \in T(v)\} \tag{4}$$

as the transmission attempts assignment for nodes in tree $T(v)$, that is, for any $u \in T(v)$, its maximal number of transmission attempts is k_u.

Next, we deduce the end-to-end delay of data aggregation of a query under the transmission attempts assignment. Let $D(H_{T(v)})$ be the maximal end-to-end delay from leaf nodes in the subtree $T(v)$ to v. In order to aggregate packets, each node can transmit only after all its children transmit packets to it. Thus $D(H_{T(v)})$ is recursively defined as

$$D(H_{T(v)}) = \begin{cases} \max\limits_{u \in c(v)}(D(H_{T(u)}) + d(u, k_u)), & \text{if } v \text{ is a non-leaf node,} \\ 0, & \text{if } v \text{ is a leaf node,} \end{cases} \tag{5}$$

where $c(v)$ is the set of v's children. Following this definition, the maximal delay for leaf nodes to have their data to be aggregated and received by the sink v_0 is $D(H_{T(v_0)})$.

Then, we analyze the amount of information that can be collected under the transmission attempts assignment. We normalize the amount of information generated at each sensor node as one. The expected information aggregated

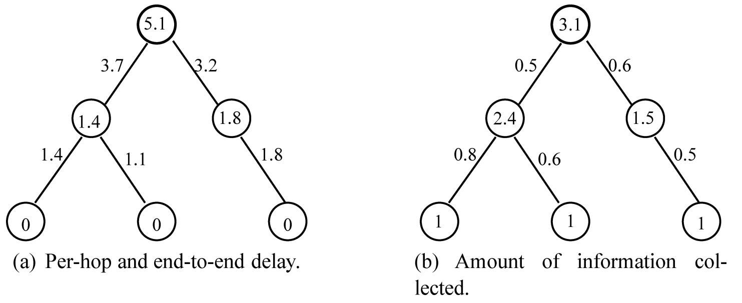
5.1
3.7
3.2
1.4
1.8
1.4
1.1
1.8
0
0
0
(a) Per-hop and end-to-end delay.
3.1
0.5
0.6
2.4
1.5
0.8
0.6
0.5
1
1
1
(b) Amount of information collected.

Fig. 1. Delay and information collected in a data aggregation tree.

at each nonleaf node v is equal to its own generated information plus the successfully received information from all its children. Let $I(H_{T(v)})$ denote the expected amount of information aggregated (and collected) at node v, which is recursively defined as

$$I(H_{T(v)}) = \begin{cases} 1 + \sum\limits_{u \in c(v)} p(u, k_u) I(H_{T(u)}), & \text{if } v \text{ is a non-leaf node,} \\ 1, & \text{if } v \text{ is a leaf node.} \end{cases} \tag{6}$$

Following this definition, the amount of information that is successfully received by the sink node v_0 is $I(H_{T(v_0)})$.

Examples of the definition of the delay and amount of information collection are shown in Figure 1. In Figure 1(a), the number beside each link represents per-hop delay, and the number in each circle represents the maximal end-to-end delay. In Figure 1(b), the number beside each link represents the probability to transmit a packet successfully, and the number in each circle represents the amount of information collection at that node. Taking the node at the root as an example, its maximal end-to-end delay is $5.1 = max(1.4 + 3.7, 1.8 + 3.2)$, and its amount of information collection is $3.1 = 1 + 0.5 \times 2.4 + 0.6 \times 1.5$.

From the definitions of Equations (1) and (3), we can see that if we assign more transmission attempts to a node, it will have a higher success probability to transmit its aggregated data back to the sink, but it will incur a higher delay, which would make the end-to-end delay exceeding the required bound. Our main task is to find the optimal assignment of the maximal allowed transmission attempts for each node in the system. The problem can be formally formulated as follows: given an aggregation tree of nodes $v_i \in V$ rooted at sink v_0, users issue queries to the sink periodically and the information must be collected to the sink with a delay constraint D_0. Our objective is to find a transmission attempts assignment vector $H(T(v_0))$, such that the expected amount of information received at the sink, $I(H_{T(v_0)})$, is maximized, subject to the constraint $D(H_{T(v_0)}) \le D_0$. That is:

$$\max_{H(T(v_0))} I(H_{T(v_0)})$$

$$\text{such that} \begin{cases} D(H_{T(v_0)}) \le D_0, \\ 1 \le k_v \le M, \; \textit{for all } (v, k_v) \in H_{T(v_0)}, \end{cases} \tag{7}$$

and k_v is an integer.

3. OPTIMAL SOLUTION

In this section, we will analyze the problem, and present the algorithm to find the optimal solution.

3.1 Mathematical Analysis

The formulation in Equation (7) consists of an objective function and a set of constraints and its optimal solution cannot be easily computed. We now transform this formulation into a dynamic programming format, which can be solved by using dynamic programming techniques. We first define a sign function:

$$sign(x, y) = \begin{cases} 1, & \text{if } x \geq y, \\ 0, & \text{if } x < y. \end{cases} \tag{8}$$

Let δ denote the delay bound at a node. It is the total amount of time that is allowed to use to collect the data from this node's descendants up to itself. We define a function $G(\delta, H_{T(v)})$ to combine the delay constraint in Equation (7) with its objective function by using the defined $sign(x, y)$, as the following:

$$G(\delta, H_{T(v)}) = p(v, k_v) I(H_{T(v)}) sign(\delta, D(H_{T(v)}) + d(v, k_v)). \tag{9}$$

It is the amount of information that is expected to be received at the parent of v from v when the end-to-end delay from v's descendants to v's parent is not exceeding δ; otherwise it equals zero.

Then, we have the following:

LEMMA 1. *Solving Equation (7) is equivalent to solve*

$$\max_{H_{T(v_0)}} \{G(D_0, H_{T(v_0)}) | G(D_0, H_{T(v_0)}) > 0\}$$
$$\textit{such that } 1 \leq k_v \leq M, \textit{ for all } (v, k_v) \in H_{T(v_0)}, \tag{10}$$

where k_v is an integer.

PROOF. First of all, $p(v_0, k_{v_0}) = 1$ and $d(v_0, k_{v_0}) = 0$ according to definition. Thus $G(D_0, H_{T(v_0)}) = I(H_{T(v_0)}) sign(D_0, D(H_{T(v_0)}))$. If $\forall H_{T(v_0)}$, $D_0 < D(H_{T(v_0)})$,

$$G(D_0, H_{T(v_0)}) = I(H_{T(v_0)}) sign(D_0, D(H_{T(v_0)})) = 0.$$

Both Equations (7) and (10) have no solution.

If $\exists H^{\star}_{T(v_0)}$, such that $D_0 \geq D(H^{\star}_{T(v_0)})$,

$$\max_{H_{T(v_0)}} G(D_0, H_{T(v_0)}) = \max_{D_0 \geq D(H_{T(v_0)})} I(H_{T(v_0)}) > 0.$$

The optimal solutions of Equations (7) and (10) are the same. □

Now we try to express $G(\delta, H_{T(v)})$ as a dynamic programming function. We consider two separate cases: (1) the remaining delay at v, that is, δ, does not permit a feasible solution, (Lemma 2); and (2) there exists a solution at v (Lemmas 3 and 4). When there is a solution, we further consider two cases: (1) v is a leaf node, that is, $c(v) = \emptyset$ (Lemma 3); and (2) v is a non-leaf node, that is, $c(v) \neq \emptyset$ (Lemma 4).

First of all, we show the sufficient condition that Equation (10) does not have a solution.

LEMMA 2. *If $d(v,1) > \delta$, then $G(\delta, H_{T(v)}) = 0$.*

PROOF. According to Equation (3), $d(v, k_v)$ increases with k_v. Therefore

$$\begin{aligned} d(v,1) > \delta &\Leftrightarrow \forall k_v, d(v,k_v) > \delta \Leftrightarrow D(H_{T(v)}) + d(v,k_v) \geq d(v,k_v) > \delta \\ &\Leftrightarrow sign(\delta, D(H_{T(v)}) + d(v,k_v)) = 0. \end{aligned}$$

Thus

$$G(\delta, H_{T(v)}) = p(v,k_v) I(H_{T(v)}) sign(\delta, D(H_{T(v)}) + d(v,k_v)) = 0. \quad \square$$

Next, we show the optimal solution at the leaf node when there exists an optimal solution.

LEMMA 3. *If $c(v) = \emptyset$ and $d(v,1) \leq \delta$, then*

$$\arg\max_{H_{T(v)}} G(\delta, H_{T(v)}) = \{(v,h)\}, \tag{11}$$

such that

$$h = \arg\max_{k_v}\{p(v,k_v) | d(v,k_v) \leq \delta, 1 \leq k_v \leq M\}, \tag{12}$$

where $\arg\max_x\{f(x)\}$ is the value of x such that $f(x)$ attains its maximal value.

PROOF. Since $c(v) = \emptyset$, then $I(H_{T(v)}) = 1$, $D(H_{T(v)}) = 0$, and $H_{T(v)} = \{v, k_v\}$. Thus

$$\begin{aligned} G(\delta, H_{T(v)}) &= p(v,k_v) I(H_{T(v)}) sign(\delta, D(H_{T(v)}) + d(v,k_v)) \\ &= p(v,k_v) sign(\delta, d(v,k_v)) \\ &= \begin{cases} p(v,k_v), & d(v,k_v) \leq \delta, \\ 0, & d(v,k_v) > \delta. \end{cases} \end{aligned}$$

Since $d(v,1) \leq \delta$, $G(\delta, H_{T(v)})$ is maximized for a $H^{\star}_{T(v)}$, which maximizes $p(v,k_v)$, while guarantees that $d(v,k_v) \leq \delta$. That is,

$$\begin{aligned} H^{\star}_{T(v)} &= \{(v,h)\}, \\ h &= \arg\max_{k_v}\{p(v,k_v) | d(v,k_v) \leq \delta, 1 \leq k_v \leq M\}. \end{aligned} \quad \square$$

Now we describe how to express $\max_{H_{T(v)}} G(\delta, H_{T(v)})$ recursively at the nonleaf node.

LEMMA 4. *If $c(v) \neq \emptyset$, then*

$$\max_{H_{T(v)}} G(\delta, H_{T(v)}) = \max_{1 \leq k_v \leq M} \left\{ p(v,k_v) \cdot \left(1 + \sum_{u \in c(v)} \max_{H_{T(u)}} G(\delta - d(v,k_v), H_{T(u)})\right) \cdot \left[\prod_{u \in c(v)} sign\left(\delta - d(v,k_v), D\left(\arg\max_{H_{T(u)}} G(\delta - d(v,k_v), H_{T(u)})\right) + d(u,k_u)\right)\right]\right\}. \tag{13}$$

PROOF.

$$G(\delta, H_{T(v)}) = p(v,k_v)I(H_{T(v)})sign(\delta, D(H_{T(v)}) + d(v,k_v))$$
$$= p(v,k_v)\cdot\left(1+\sum_{u\in c(v)} p(u,k_u)I(H_{T(u)})\right) sign\left(\delta, d(v,k_v) + \max_{u\in c(v)}(D(H_{T(u)}) + d(u,k_u))\right)$$
$$= p(v,k_v)\cdot\left(1+\sum_{u\in c(v)} G(\delta - d(v,k_v), H_{T(u)})\right)$$
$$\cdot \prod_{u\in c(v)} sign(\delta - d(v,k_v), D(H_{T(u)}) + d(u,k_u)).$$

Therefore

$$\max_{H_{T(v)}} G(\delta, H_{T(v)}) = \max_{1\le k_v\le M}\left\{p(v,k_v)\cdot\left(1+\sum_{u\in c(v)}\max_{H_{T(u)}} G(\delta - d(v,k_v), H_{T(u)})\right)\cdot\right.$$
$$\left.\left[\prod_{u\in c(v)} sign\left(\delta - d(v,k_v), D\left(\arg\max_{H_{T(u)}} G(\delta - d(v,k_v), H_{T(u)})\right) + d(u,k_u)\right)\right]\right\}. \quad \square \tag{14}$$

Based on the above lemmas, we can formulate the problem as a dynamic programming function by the following theorem.

THEOREM 1. *The solution to problem in formulation (7) can be formulated as the following:*

—*Case* $c(v) = \emptyset$:

$$\arg\max_{H_{T(v)}}\{G(\delta, H_{T(v)})|G(\delta, H_{T(v)}) > 0\} = \begin{cases} \emptyset, & d(v,1) > \delta, \\ \{(v,h)\}, & otherwise, \end{cases}$$

where $h = \arg\max_{k_v}\{p(v,k_v)|d(v,k_v) \le \delta, 1 \le k_v \le M\}$.

—*Case* $c(v) \ne \emptyset$:
If there exists

$$k_v^{max} =$$
$$\max_{1\le k_v\le M}\left\{k_v | \prod_{u\in c(v)} sign\left(\delta - d(v,k_v), D\left(\arg\max_{H_{T(u)}} G(\delta - d(v,k_v), H_{T(u)})\right) + d(u,k_u)\right) > 0\right\},$$

then

$$\arg\max_{H_{T(v)}}\left\{G(\delta, H_{T(v)})|G(\delta, H_{T(v)}) > 0\right\} = \{(v,h)\}$$
$$\cup \bigcup_{u\in c(v)} \arg\max_{H_{T(u)}} G(\delta - d(v,h), H_{T(u)}),$$

where

$$h = \arg \max_{1 \le k_v \le k_v^{max}} \left\{ p(v, k_v) \left(1 + \sum_{u \in c(v)} \max_{H_{T(u)}} G(\delta - d(v, k_v), H_{T(u)}) \right) \right\}.$$

Else,

$$\arg \max_{H_{T(v)}} \{G(\delta, H_{T(v)}) | G(\delta, H_{T(v)}) > 0\} = \emptyset.$$

Proof. This is the direct result from Lemmas 1, 2, 3, and 4. □

3.2 Dynamic Programming Algorithm

Based on the Theorem 1, we present the dynamic programming algorithm to find the optimal solution to our defined problem.

The algorithm is shown in Algorithm 1. The basis of the algorithm is the recursive function $G_{max}(v, \delta, H_{T(v)}, I(H_{T(v)}))$. This function computes the amount of the information collected at v, $I(H_{T(v)})$, for the optimal assignment of transmission attempts $H_{T(v)}$, given the subtree $T(v)$ and the end-to-end delay bound of this subtree δ. When v is a leaf node, the optimal assignment of transmission attempts is simply $\{(v, h)\}$, where h is the largest number that still makes the transmission delay from v to its parent, $d(v, h)$, not exceeding δ. The success probability for v to deliver the data packet to its parent, $p(v, h)$, is the maximal in this case. When v has children, the algorithm tries all possible transmission attempts k_v and searches all v's children to find the best assignment that will return the optimal amount of information for subtree $T(v)$ by recursively calling $G_{max}(u, \delta - d(v, k_v), H_{T(u)}, I(H_{T(u)}))$, where u is a child of v. We introduce a variable G_{tmp} to record the so far obtained maximal amount of information that is expected to be received by v's parent from v. The algorithm searches the best value of k_v from 1 to M. For each given k_v, if the $H_{T(u)}$ returned from v's child is empty, indicating that the current value of k_v is too large to meet the delay constraint, it returns; otherwise it records the transmission attempts of this child and the amount of information expected to be received from this child. For a given k_v, after searching subtree $T(v)$, if the amount of information that is expected to be received by v's parent from v is larger than so far obtained G_{tmp}, then G_{tmp} is updated and $H_{T(v)}$ and $I(H_{T(v)})$ are recorded. The algorithm starts the search by calling $G_{max}(v_0, D_0, H_{T(v_0)}, I(H_{T(v_0)}))$, where v_0 is the sink and D_0 is the end-to-end delay bound for data aggregation.

3.3 Time Complexity of the Optimal Algorithm

In this subsection we analyze the time complexity of the dynamic programming algorithm, Algorithm 1.

Theorem 2. *Algorithm 1 has time complexity $O(NM^L)$, where N is the number of nodes, M the upper limit of transmission attempts, and L the height of the data aggregation tree.*

Proof. We denote the average number of children of each node in the tree as K, the number of nodes in the tree as N, and the upper limit of transmissions attempts as M.

Algorithm 1 The algorithm of optimal solution by dynamic programming.

```
Main{
$G_{max}(v_0, D_0, H_{T(v_0)}, I(H_{T(v_0)}))$;    /*Start from sink $v_0$*/
}

$\mathbf{G_{max}}(\mathbf{v}, \delta, \mathbf{H_{T(v)}}, \mathbf{I(H_{T(v)})})${    /*Recursive function definition*/
if $c(v) = \emptyset$ then    /*Leaf node*/
  if $\delta < d(v, 1)$ then    /*No feasible solution*/
    $H_{T(v)} = \emptyset$; $I(H_{T(v)}) = 0$;
  else
    $h = \arg\max_{k_v}\{p(v, k_v)|d(v, k_v) \leq \delta, 1 \leq k_v \leq M\}$;
    $H_{T(v)} = \{(v, h)\}$; $I(H_{T(v)}) = 1$;
  end if
else    /*Non-leaf node*/
  $G_{tmp} = 0$;
  for $k_v$ = 1 to $M$ do    /*Search for all possible $k_v$*/
    $H_{tmp} = \emptyset$; $I_{tmp} = 0$; $H_{T(v)} = \emptyset$; $I(H_{T(v)}) = 0$;
    for all $u \in c(v)$ do    /*Search for all children of $v$*/
      $G_{max}(u, \delta - d(v, k_v), H_{T(u)}, I(H_{T(u)}))$;    /*Recursive call*/
      if $H_{T(u)} = \emptyset$ then    /*No feasible solution*/
        return;
      else
        $H_{tmp} = H_{tmp} \cup H_{T(u)}$; $I_{tmp} = I_{tmp} + p(u, k_u)I(H_{T(u)})$;
      end if
    end for
    $H_{tmp} = H_{tmp} \cup \{(v, k_v)\}$; $I_{tmp} = I_{tmp} + 1$;
    if $p(v, k_v)I_{tmp} > G_{tmp}$ then
      $G_{tmp} = p(v, k_v)I_{tmp}$; $H_{T(v)} = H_{tmp}$; $I(H_{T(v)}) = I_{tmp}$;
    end if
  end for
end if
}
```

Let L be the height of the data aggregation tree. We have

$$\frac{K^L - 1}{K - 1} = N. \tag{15}$$

Hence

$$L = \frac{\lg(N(K-1)+1)}{\lg K}. \tag{16}$$

We define the computation cost of a node at layer i of the aggregation tree as $F(i)$, where the root is at layer L and the leaf node is at layer 1. According to the Algorithm 1, we have

$$F(i+1) = MK \cdot F(i) + \epsilon(i+1), \tag{17}$$

where $\epsilon(i+1) = O(1)$. This is because each node needs to check M different transmission attempts, from the assignments of K children, to report an optimal assignment to its parents.

We have $F(1) = M$, as a leaf node needs to check M times of the maximal number of transmission attempts.

Hence totally the time complexity of the dynamic programming algorithm is

$$O(M(MK)^{L-1}) = O(K^{L-1} \cdot M^L) = O(NM^L). \tag{18}$$

□

4. DISTRIBUTED HEURISTIC ALGORITHM

Although Algorithm 1 can obtain the optimal solution, it has some drawbacks: (1) the optimal algorithm has an time complexity of $O(NM^L)$, which increases quickly as the increase of the height of the aggregation tree L. It may not be able to compute the optimal solution for large scale sensor networks. (2) The algorithm works in a centralized fashion. This requires a central node that has the global information about the entire network. This is not practical for large-scale sensor networks.

We propose a distributed algorithm that has the similar performance as the optimal one, with much less time complexity. From the theoretical analysis, we observe that the maximal number of transmission attempts assigned to a node is affected by two major factors: the amount of information aggregated at this node and the number of other nodes that interfere with this node. The more amount of information aggregated at a node and to be forwarded to its parent, the larger the maximal number of transmission attempts that should be assigned to this node, to avoid a large amount of information loss. The more nodes that a node interferes with, the smaller maximal number of transmission attempts we need to assign it, because this node has a larger per-attempt delay. However, when the sensor nodes are uniformly distributed, the contention level of nodes (i.e., the number of nodes that it interferes with) are more or less the same. This leads to an observation that the transmission attempts assignment is largely dominated by the amount of information aggregated at the nodes. This observation is also confirmed by the experimental results of the optimal solution. From the optimal results computed by Algorithm 1, we found the nodes that are closer to the sink usually have a larger maximal number of transmission attempts, because these nodes have aggregated more information from their descendants. Based on this observation, we propose the distributed *greedy* algorithm.

The details of the greedy algorithm are shown in Algorithm 2. First of all, we need to find out whether there exists a feasible solution to meet the end-to-end delay constraint. As we know, when the number of transmission attempts of each node is set to 1, the end-to-end delay becomes the minimal. Let $H^{min}_{T(v_0)}$ denote the assignment that the maximal number of transmission attempts of each node in tree $T(v_0)$ is assigned to 1. That is:

$$H^{min}_{T(v_0)} = \{(v_i, 1) | v_i \in T(v_0)\}, \tag{19}$$

where v_0 is the sink. Since the delay for one transmission attempt of a node v, that is, $d(v, 1)$, depends on the interference situation of v, which is different from node to node, the sink can simply collect this delay information from nodes, without computing it by itself. Therefore, there is no need for the sink to have the global location information of all nodes in the network. Then, the

Algorithm 2 Greedy transmission attempts assignment algorithm.

```
Sink node $v_0$ do{
Collect $d(v, 1)$ from all $v$, $v \in T(v_0)$;
Compute $H^{min}_{T(v_0)}$;
$\Delta = (D_0 - D(H^{min}_{T(v_0)}))$;
if $\Delta \geq 0$ then    /*Feasible solution exist*/
  for all $u \in c(v_0)$ do
    delay_update_request($u$, $\Delta$);    /*Send update request to children*/
  end for
else    /*No feasible solution*/
  report no feasible solution;
end if
}

For each node $v$, upon receiving delay_update_request, do{
$k_v = M$;
while $d(v, k_v) - d(v, 1) > \Delta$ do
  $k_v = k_v - 1$;
end while
for all $u \in c(v)$ do
  delay_update_request($u$, $\Delta-(d(v, k_v)-d(v, 1))$);    /*Send delay surplus to children*/
end for
}
```

sink computes the delay surplus. Let $D(H^{min}_{T(v_0)})$ be the end-to-end delay in the assignment of $H^{min}_{T(v_0)}$. When $D_0 < D(H^{min}_{T(v_0)})$, there is no feasible solution. When $D_0 \geq D(H^{min}_{T(v_0)})$, we define the delay surplus Δ as

$$\Delta = \left(D_0 - D\left(H^{min}_{T(v_0)}\right)\right). \tag{20}$$

The delay surplus Δ indicates the amount of extra delay that can be allocated to sensor nodes under the condition that every node has already been given one transmission attempt. The greedy algorithm greedily assigns the transmission attempts from the top down by allocating extra delay budget hop by hop. The basic idea is to assign more transmission attempts to the nodes that are close to the sink, because they aggregate more information in their data packets. The sink sends a delay update request to all its children with the surplus Δ. Upon receiving this delay update request with Δ, each node v increases its maximal number of transmission attempts by using Δ as much as possible, or its maximal number of transmission attempts reach the upper limit M. Suppose node v increases its transmission attempts to k_v, $k_v \leq M$. It must make sure $(d(v, k_v) - d(v, 1)) \leq \Delta$. Since the success probability of packet delivery $p(v, k_v)$ increases as the increase of k_v, $p(v, k_v)$ reaches its highest possible value when $d(v, k_v)$ cannot increase anymore. After revising its maximal number of transmission attempts, node v sends an update request with the remaining delay surplus $\Delta - (d(v, k_v) - d(v, 1))$ to all its children. This operation is repeated node by node from top-down along the aggregation tree, until reaching the leaf nodes. The algorithm terminates after all nodes receive the message requesting revising the number of transmission attempts. Therefore, the number of messages used in the algorithm is $O(N)$.

To evaluate the performance of the optimal algorithm and the greedy algorithm, we introduce a benchmark method, *even-distribution*. The even-distribution method simply computes the delay surplus as in the greedy algorithm, and divides the delay surplus by the height of the aggregation tree. The delay surplus is evenly distributed to nodes in the tree. Each node increases its transmission attempts to the largest possible value to use up the delay surplus allocated to it. Note that, since nodes have different delays for one transmission attempt, they have a different number of transmission attempts, even though they are allocated with the same amount of delay surplus.

5. SIMULATION RESULTS

In this section, we evaluate the performance of the optimal algorithm (dynamic programming), greedy algorithm, and even-distribution method (EVEN), under various system parameters such as, end-to-end delay bounds, network topologies, and heights of aggregation trees. The simulation is done by a C++ simulator. Since our algorithm works on delay bound assignment, rather than the modification of MAC protocol, this simulator is enough for us to understand how these algorithms perform.

5.1 Simulation Configurations

The end-to-end delay and amount of information collected at the sink is obtained by the mathematical model described in Section 2. The queries are executed periodically, and the corresponding data generating rate of each node, λ, is set to 20 packets/s. The data size is fixed as 128 bytes. In the MAC layer, the size of contention window, w, is fixed as 32, and the upper limit of the maximal number of transmission attempts, M, is set to 4. The values M is the default settings in IEEE 802.11 systems, and w is the default setting of the minimal contention window.

We consider an omnidirectional signal attenuation rate in this simulation, that is, the signal attenuation rate $A(u, v)$ depends only on the distance between u and v. The transmission and interference model with such a configuration is a disk model. Each node has a fixed transmission distance r_t. Each node interferes with another one, when their distance is less than qr_t, where q is the ratio of interference distance to the transmission distance. The interference area of any node is the disk centered at this node, with radius qr_t. We set q as 2 in this simulation. This disk model may not perfectly illustrate the interference distribution in practice, because the practical interference area is irregular, as reported in Zhao and Govindan [2003], Woo et al. [2003], and Zhou et al. [2004, 2005, 2006]. However, we think the simulation result under this disk model is still representative, due to the following reasons: (1) according to the investigation in Zhou et al. [2004, 2006], the radio irregularity has less impact on the MAC layer, when the MAC layer protocol is IEEE 802.11 and the upper layer routing protocol can deal with radio irregularity. In this article, we focus on MAC layer parameter setting, rather than routing tree construction. It is reasonable to assume that a routing algorithm that mitigates radio irregularity is adopted, such as AODV [Perkins and Royer 1999], DSR [Johnson and Maltz 1996], and symmetric geographic forwarding [Zhou et al. 2004; Zhou et al. 2006]. In addition, since we employ an IEEE802.11-like MAC protocol, the impact of radio irregularity can be regarded as marginal. (2) The interference level in practice varies from scenarios to scenarios. It is not suitable for us to draw general conclusions from a particular interference distribution. The disk model is preferred, as it is a good simple model to illustrate the insight of how our algorithms perform in general cases.

We consider two kinds of networks in the simulations: grid networks and random networks. The grid networks have uniform pattern of node distributions and regular topologies. Thus the network parameters, such as number of end-to-end hops, tree structure, and number of nodes in the interference range, are deterministic. By simulating the algorithms in grid networks, we can get a clear understanding of the relationship between different network parameters and the performance of algorithms. The random networks have random node distributions, and the topologies are created from random connections, which are closer to the scenario of practical sensor networks.

5.2 Grid Networks

The grid network consists of N nodes, where the number of rows is same as the number of columns. The distance between two neighbors in the same row or the same column is set as a constant a. The transmission distance is $1.75a$, such that a node can directly communicate with its neighbors in the same row or column, or its diagonal neighbors. The interference distance is twice as much of the transmission distance. The sink is placed at the top-left corner of the area. We will study how the amount of information collected is affected by the network parameters such as end-to-end delay bound, tree height, and size of the set of interfering nodes. Throughout the simulation, we use the information collection ratio as a metric to represent the amount of information collected at the sink, which is defined as the amount of information collected at the sink divided by the number of nodes in the network.

The network in this batch of simulations is an 8×8 grid network. We first study a special topology of the aggregation tree that all leaf nodes have the same number of hops to the sink. The topology of this aggregation tree is shown in Figure 2. Intuitively, when a path from root to leaf node is longer, we need to assign a fewer number of transmission attempts to nodes on this path, to avoid the end-to-end delay exceeding the delay bound. For the tree topology in Figure 2, the number of attempts allocated to a node is independent of which path it is in, because all tree paths (to leaf nodes) have the same length. We can focus on how the interference factor affects the number of attempts assigned to nodes.

Figure 3 shows the information collection ratio versus end-to-end delay bounds. The end-to-end delay bound is in the unit of $D(H_{T(v_0)}^{min})$, which is the longest delay from any leaf node to the sink along the aggregation tree when every node has only one transmission attempt. For the aggregation tree shown in Figure 2, $D(H_{T(v_0)}^{min})$ is 1.76×10^{-2} s. There will be no feasible solution for

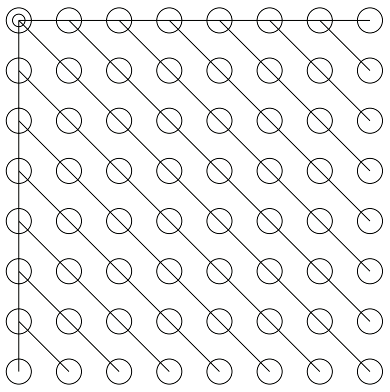

Fig. 2. Topology of 8 × 8 grid networks of same end-to-end hops between leafs and the sink.

any delay bound smaller than this value. When the end-to-end delay bound is larger than 1.3, the optimal solution assigns M, the upper limit of transmission attempts, to all nodes. From Figure 3, we observe that the performance of the greedy algorithm is almost the same as that of the optimal one, and is significantly better than that of the even-distribution method. For both the optimal and greedy algorithms, the information collection ratio becomes stable after the end-to-end delay bound is greater than $1.3D(H_{T(v_0)}^{min})$. Table I shows the maximal number of transmission attempts assigned by optimal and greedy algorithms to each node for the tree in Figure 2, when the end-to-end delay bound is $1.03D(H_{T(v_0)}^{min})$. In this case, as shown in the table, the assignment of transmission attempts done by the optimal algorithm is identical to the assignment done by the greedy algorithm. Notice that the even-distribution method assigns one transmission attempt for every node in this case. From the number of transmission attempts in Table I, we find that (1) the optimal solution assigns more transmission attempts to the nodes closer to the sink (among nodes with similar interference levels). This is because the data packets transmitted by the nodes closer to the sink contain more information. An example is that the node at location (7, 7) is assigned more transmission attempts than the node at location (8, 8), because the former node carriers nearly two times of information of the latter node. (2) The optimal solution assigns the nodes with more interference fewer transmission attempts (among nodes with similar hops to the sink). From Table I, we can see it assigns the nodes around the center region in the grid network only one transmission attempt. The nodes in the center region have more nodes interfering with them. The nodes at the edges of the grid have lighter interference and they are assigned four transmission attempts (except the node in the right-bottom corner). This is because the per-hop delay for a node with heavier interference is much larger than a node with lighter interference (see Equation (3)), and this delay increases dramatically as the number of transmission attempts increases. To maximize the overall information collection ratio, the optimal solution assigns fewer transmission attempts to these nodes having heavier interference, so that it leaves more transmission attempts to the nodes having lighter interference. (3) The

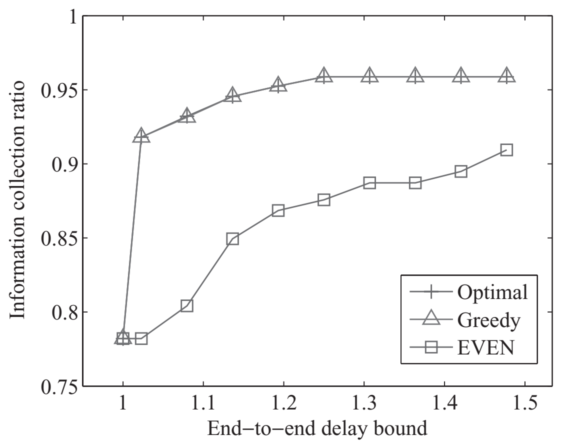

Fig. 3. Information collection ratio versus end-to-end delay bounds for the aggregation tree in Figure 2.

Table I. Transmission Attempts for Nodes in Figure 2.

Max. # of attempts		Columns							
		1	2	3	4	5	6	7	8
Rows	1	—	4	4	4	4	4	4	4
	2	4	4	4	4	4	4	4	4
	3	4	4	1	1	1	4	4	4
	4	4	4	1	1	1	1	4	4
	5	4	4	1	1	1	1	4	4
	6	4	4	4	1	1	1	4	4
	7	4	4	4	4	4	4	4	4
	8	4	4	4	4	4	4	4	1

optimal solution assigns the nodes on a path of a higher contention level fewer transmission attempts. An example is that, although the node at location (8, 8) and the node at location (8, 7) have similar interference levels and same hops to the sink, the former nodes are assigned less transmission attempts. The reason is as follows: for a path of a high contention level, its end-to-end delay (from the sink to the leaf node) is large; so that the optimal assignment will not assign too many transmission attempts to nodes on this path, to avoid the situation where the delay exceeds the bound.

Next, we study the aggregation tree where leaf nodes have different numbers of hops to the sink. The topology of this aggregation tree is shown in Figure 4. By studying this tree topology, we can see that the number of transmission attempts assigned to a node not only depends on the node interference, but also on the length of the tree path that this node is in.

Figure 5 shows the information collection ratio versus end-to-end delay bounds. The end-to-end delay bound is in the unit of $D(H^{min}_{T(v_0)})$, which is 2.71×10^{-2} s for the aggregation tree in Figure 4. Similarly to the observation from Figure 3, we observe that the performance of the greedy algorithm is almost the same as the optimal one, and is significantly better than for the

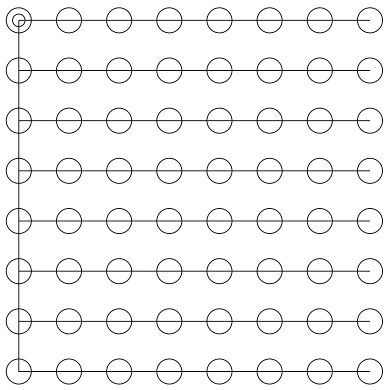

Fig. 4. Topology of 8 × 8 grid networks of different end-to-end hops between leafs and the sink.

even-distribution method. Table II shows the maximal number of transmission attempts assigned by optimal and greedy algorithms to each node in the tree in Figure 4, when the end-to-end delay bound is $1.03D(H^{min}_{T(v_0)})$. In the table, when there is only one number in a table cell, it means both optimal and greedy algorithms assign the same number of attempts to the node; otherwise the numbers listed in brackets are computed by the greedy algorithm. We can see, in most cases, the optimal algorithm and the greedy algorithm assign the same number of transmission attempts to sensor nodes. When the nodes' interference levels are similar, the optimal assignment gives more transmission attempts to the nodes closer to the sink. For example, the nodes in row 3 have similar interference levels as the nodes in row 6, but are assigned more transmission attempts. When the nodes' numbers of hops to the sink are similar, the optimal algorithm assigns fewer attempts to the nodes with heavier interference. For example, the nodes in the center are assigned fewer transmission attempts. Although the greedy algorithm assigns different transmission attempts to some nodes, compared with the optimal one, the difference in the information collection ratio is almost negligible. The even-distribution method evenly distributes the delay surplus to all the tree nodes. It converts the per-hop surplus into a number of attempts, which may be different for the different nodes due to different contention levels. It allocates four transmission attempts to the three corner nodes (except the top-left corner) and one attempt to all the other nodes. This is because the corner nodes have the lightest interference in the network and with the same amount of delay it can be converted into a more transmission attempts.

Finally, we study how the number of nodes in the network affects the information collection ratio. We expand the grid network in Figure 2 by adding more rows and columns. The size of the grid network increases from 8 × 8 to 12 × 12. The aggregation tree topology is the same as that in Figure 2.

During the simulation, each time we increase the network size, we compute $D(H^{min}_{T(v_0)})$ and set the end-to-end delay bound to $1.02 \times D(H^{min}_{T(v_0)})$. Figure 6 shows the information collection ratio versus the size of the network. In addition to the same result we observed before that the greedy algorithm performs almost

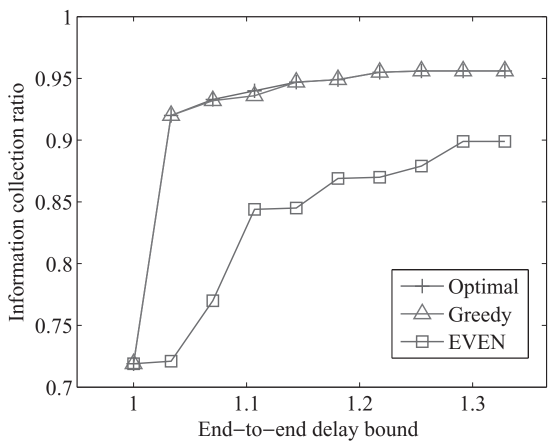

Fig. 5. Information collection ratio versus end-to-end delay bounds for the aggregation tree in Figure 4.

Table II. Transmission Attempts for Nodes in Figure 4.

Max. # of attempts		Columns							
		1	2	3	4	5	6	7	8
Rows	1	-	4	4	4	4	4	4	4
	2	4	4	4	4	4	4	4	4
	3	4	4	4	2	2	4	4	4
	4	4	4	2	1	1	1	1	4
	5	4	1	1	1	1	1	1	4
	6	4	2 (4)	2	1	1	1	1	2 (1)
	7	4	4	4	4	2 (4)	2 (4)	4 (1)	4
	8	4	4	4	4	4	4	4	4

the same as the optimal one, we also find that (1) as the network size increases (i.e., as the tree height increases), the information collection ratio decreases. This is because as the tree heights increase, there is a higher end-to-end packet loss rate. (2) The performance gap between the optimal algorithm and the even-distribution method widens as the network size increases. This is because in the even-distribution method the packet loss rate at nodes close to the sink is higher than in the optimal solution, and it results in more information loss as more packets are aggregated in a larger network. The observation from Figure 6 indicates that delay-constrained multihop wireless sensor networks may have low information collection ratios, when the number of end-to-end hops is large. There are several approaches to address this issue. One is placing multiple sinks into networks, so that the numbers of hops from sensors to sinks decrease. Another approach is opportunistic routing [Biswas and Morris 2004], or opportunistic data aggregation [Chen and Shin 2008]. Since the wireless channel is a broadcast medium naturally, information can be sent by sensors to multiple receivers in a broadcast way, and forwarded to the sink in multiple paths. Since multiple copies of the information are transmitted, the network with opportunistic routing or data aggregation can tolerate more data losses.

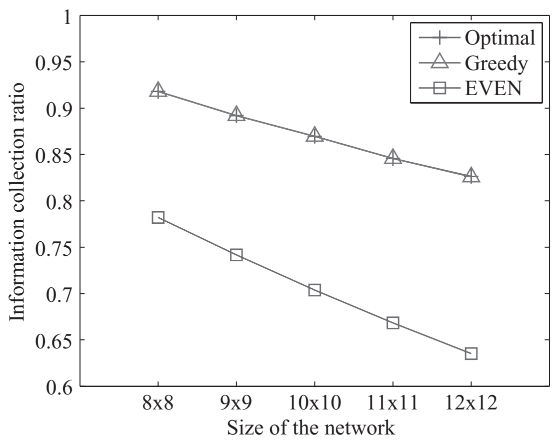

Fig. 6. Information collection ratio in grid networks versus size of the network.

5.3 Random Networks

For the simulation of random networks, we randomly distribute sensor nodes in a $100 \times 100\text{-m}^2$ square region. The sink is placed at the center of this area. The transmission distance of nodes is set to 20m. The interference distance is twice as much as the transmission distance. For each simulation, we compute a network based on the location of nodes and transmission distance. We simply take the shortest path tree as data aggregation tree for the generated network.

We first study how the number of nodes in the network affects the performance of the algorithms. We start from 60 nodes in the network, and gradually increase the number up to 100. When the number of nodes is below 60, the networks generated are often disconnected. During this simulation, we always set the end-to-end delay bound as $1.1 \times D(H^{min}_{T(v_0)})$. Notice that $D(H^{min}_{T(v_0)})$ increases as the number of nodes in the network increases; thus it is fairer to evaluate the information collection ratio with a delay bound proportional to $D(H^{min}_{T(v_0)})$. Figure 7 shows the information collection ratio versus the number of nodes in the network. The greedy algorithm performs almost the same as the optimal one. In addition, we observe that (1) as the of number of nodes increases, the information collection ratio decreases. This is because as the number of nodes increases (which means the node density increases), the interference level increases, resulting in a larger per-hop delay and a higher per-hop packet loss ratio. (2) The performance gap between the optimal algorithm and the even-distribution method shortens as the number of nodes increases. This is because when the number of nodes is sufficiently large, the per-hop delay to transmit a packet multiple times is so large that most nodes can only transmit packets once, in order to meet the end-to-end delay bound. The observations from Figure 7 suggest that pure CSMA/CA protocols may have a low information collection ratio for dense sensor networks, due to too many contentions. In such scenarios, some CSMA/CA variants that decrease per-hop latency, such as DB-MAC [Bacco et al. 2004], or packet losses, such as funneling-MAC [Ahn et al. 2006], will help improve the system performance.

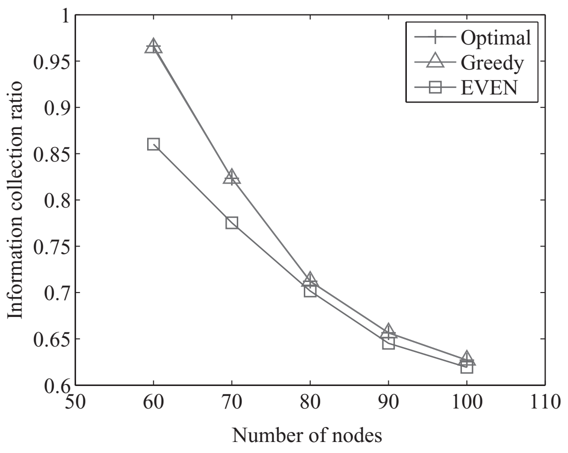

Fig. 7. Information collection ratio versus number of nodes.

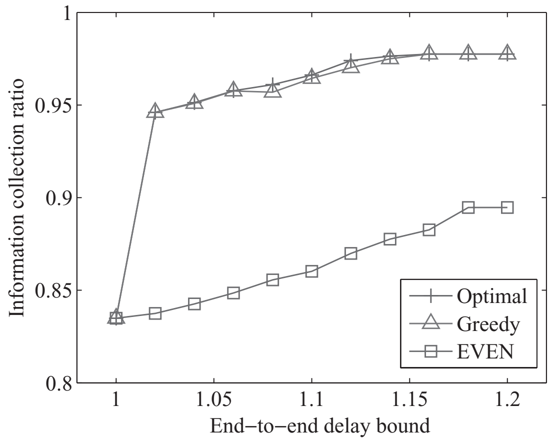

Fig. 8. Information collection ratio versus end-to-end delay bounds.

Next, we study how the end-to-end delay bounds affect the performance. In this simulation, we set the number of nodes in the network to 60. The end-to-end delay bounds are in the unit of $D(H^{min}_{T(v_0)})$, which is 1.18×10^{-2} s for this random network. Figure 8 shows the information collection ratio versus the end-to-end delay bounds. The greedy algorithm still performs almost the same as the optimal one in this scenario. We observe that, as the end-to-end delay bounds increases, the information collection ratio increases. This is because more transmission attempts are allowed. For both the optimal and greedy algorithms, the information collection ratio becomes stable after the end-to-end delay bound is greater than $1.15D(H^{min}_{T(v_0)})$, because the algorithms assign the upper limit of the maximal transmission attempts for each node in this case.

6. CONCLUSION

In this article we investigated the problem of delay constrained maximal information collection for CSMA-based wireless sensor networks. We assign the maximal number of transmission attempts to each sensor node in a data aggregation tree. The objective is to collect the maximal amount of information to the sink node while meeting the end-to-end delay constraint. We formulated the problem by using dynamic programming and proposed an optimal algorithm based on the dynamic programming formulation. Based on the analysis and observation of the optimal solutions, we proposed an efficient distributed greedy algorithm. Through simulations, we observed the following conclusions: (1) the greedy algorithm achieves almost the same performance as the optimal one, and both the greedy and optimal algorithms are significantly better than the benchmark even-distribution method. (2) To collect more information within the delay constraint, more transmission attempts should be assigned to the nodes close to the sink, and fewer transmission attempts should be assigned to the nodes with heavier interference. (3) The information collection ratio quickly reaches a stable and saturated level (around 95%) when the end-to-end delay bound is greater than 1.2 or 1.3 times the minimal required end-to-end delay (the largest end-to-end delay where each node is allowed to have only one transmission attempt).

REFERENCES

AHN, G., HONG, S. G., MILUZZO, E., CAMPBELL, A. T., AND CUOMO, F. 2006. Funneling-MAC: A localized, sink-oriented MAC for boosting fidelity in sensor networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*. 293–306.

AKYILDIZ, I., WEILIAN, S., SANKARASUBRAMANIAM, Y., AND CAYIRCI, E. 2002. A survey on sensor networks. *IEEE Commun. Mag. 40,* 8 (Aug.), 102–114.

AL-KARAKI, J., UL-MUSTAFA, R., AND KAMAL, A. 2004. Data aggregation in wireless sensor networks - exact and approximate algorithms. In *Proceedings of IEEE HPSR*. 241–245.

ALICHERRY, M., BHATIA, R., AND LI, L. E. 2005. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *Proceedings of ACM MobiCom*. 58–72.

BACCO, G., MELODIA, T., AND CUOMO, F. 2004. A MAC protocol for delay-bounded applications in wireless sensor networks. In *Proceedings of the Mediterranean Ad Hoc Networking Workshop*. 208–220.

BAROWSKI, Y., BIAZ, S., AND AGRAWAL, P. 2005. Towards the performance analysis of IEEE 802.11 in multi-hop ad-hoc networks. In *Proceedings of the IEEE WCNC*, Vol. 1. 100–106.

BIANCHI, G. 2000. Performance analysis of the IEEE802.11 distributed coordination function. *IEEE J. Select. Areas: Commun. 18*, 535–547.

BISWAS, S. AND MORRIS, R. 2004. Opportunistic routing in multi-hop wireless networks. *ACM SIGCOMM Comput. Commun. Rev. 34,* 1, 133–144.

CHEN, D. AND VARSHNEY, P. K. 2004. QoS support in wireless sensor networks: A survey. In *Proceedings of the IEEE ICWN*. 227–233.

CHEN, Z. AND SHIN, K. 2008. OPAG: Opportunistic data aggregation in wireless sensor networks. In *Proceedings of the IEEE Real-Time Systems Symposium*. 345–354.

CHIPARA, O., LU, C., AND STANKOVIC, J. 2006. Dynamic conflict-free query scheduling for wireless sensor networks. In *Proceedings of ICNP*. 321–331.

CULLER, D., ESTRIN, D., AND SRIVASTAVA, M. 2004. Overview of sensor networks. *Comput. 37,* 8, 41–49.

DU, H., HU, X., AND JIA, X. 2006. Energy efficient routing and scheduling for real-time data aggregation in WSNs. *Comput. Commun. 29,* 17, 3257–3535.

Hua, C. and Yum, T. 2008. Optimal routing and data aggregation for maximizing lifetime of wireless sensor networks. *IEEE/ACM Trans. Netw. 16,* 4, 892–903.

Huang, S., Wan, P., Vu, C., Li, Y., and Yao, F. 2007. Nearly constant approximation for data aggregation scheduling in wireless sensor networks. In *Proceedings of the IEEE INFOCOM*. 366–372.

IEEE Computer Society LAN MAN Standards Committee. 1997. IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11-1997. IEEE, Piscataway, NJ.

IEEE Computer Society LAN MAN Standards Committee, Ed. 2006. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks. IEEE Standard 802.15.4-2006. IEEE, Piscataway, NJ.

Intanagonwiwat, C., Estrin, D., Govindan, R., and Heidemann, J. 2002. Impact of network density on data aggregation in wireless sensor networks. In *Proceedings of ICDCS*. 575–578.

Intanagonwiwat, C., Govindan, R., and Estrin, D. 2000. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of ACM MobiCOM*. 56–67.

Johnson, D. B. and Maltz, D. A. 1996. *Mobile Computing*. Chapter "Dynamic Source Routing in Ad Hoc Wireless Networks." Kluwer, Norwell, MA, 153–181.

Kalpakis, K., Dasgupta, K., and Namjoshi, P. 2003. Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks. *Int. J. Comput. Telecommun. Netw. 42,* 6, 697–716.

Krishanamachari, B., Estrin, D., and Wicker, S. 2002. The impact of data aggregation in wireless sensor networks. In *Proceedings of the IEEE ICDCSW*. 575–578.

Pan, Y. and Lu, X. 2007. Energy-efficient lifetime maximization and sleeping scheduling supporting data fusion and qos in multi-sensornet. *Signal Process. 87,* 12, 2949–2964.

Perkins, C. and Royer, E. 1999. Ad-hoc on-demand distance vector (AODV) routing. In *Proceedings of the IEEE WMCSA*. 90–100.

Perrig, A., Szewczyk, R., Tygar, J., Wen, V., and Culler, D. 2002. Spins: Security protocols for sensor networks. *Wireless Netw. 8,* 5, 521–534.

Polastre, J., Hill, J., and Culler, D. 2004. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*. 95–97.

Pottie, G. and Kaiser, W. 2000. Wireless integrated network sensors. *Commun. ACM 43,* 5, 51–58.

Rajagopalan, R. and Varshney, P. 2006. Data-aggregation techniques in sensor networks: A survey. *IEEE Commun. Surv. Tutor. 8,* 4, 48–63.

Römer, K., Holger, K., and Mattern, F., Eds. 2006. *Proceedings of the 3rd European Workshop*, Lecture Notes in Computer Science, vol. 3868. Springer, Berlin, Germany, 266.

Solis, I. and Obraczka, K. 2004. The impact of timing in data aggregation for sensor networks. In *Proceedings of IEEE ICC*, Vol. 6. 3640–3645.

Tang, X. and Xu, J. 2008. Optimizing lifetime for continuous data aggregation with precision guarantees in wireless sensor networks. *IEEE/ACM Trans. Netw. 16,* 4, 904–917.

Woo, A. and Culler, D. 2001. A transmission control scheme for media access in sensor networks. In *Proceedings of ACM MobiCom*. 221–235.

Woo, A., Tong, T., and Culler, D. 2003. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proceedings of ACM SenSys*. 14–27.

Xue, Y., Cui, Y., and Nahrstedt, K. K. 2005. Maximizing lifetime for data aggregation in wireless sensor networks. *Mobile Netw. Appl. 10,* 6, 853–864.

Younis, O. and Fahmy, S. 2005. An experimental study of routing and data aggregation in sensor networks. In *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems*. 50–57.

Yu, Y., Krishnamachari, B., and Prasanna, V. 2004. Energy-latency tradeoffs for data gathering in wireless sensor networks. In *Proceedings of IEEE INFOCOM*, Vol. 1. 244–255.

Zhao, J. and Govindan, R. 2003. Understanding packet delivery performance in dense wireless sensor networks. In *Proceedings of ACM SenSys*. 1–13.

Zhou, G., He, T., Krishnamurthy, S., and Stankovic, J. A. 2004. Impact of radio irregularity on wireless sensor networks. In *Proceedings of ACM MobiSys*. 125–138.

ZHOU, G., HE, T., KRISHNAMURTHY, S., AND STANKOVIC, J. A. 2006. Models and solutions for radio irregularity in wireless sensor networks. *ACM Trans. Sensor Netw. 2,* 2, 221–262.

ZHOU, G., HE, T., STANKOVIC, J. A., AND ABDELZAHER, T. 2005. RID: radio interference detection in wireless sensor networks. In *Proceedings of IEEE INFOCOM*, Vol. 2. 891–901.

ZHU, J., PAPAVASSILIOU, S., AND YANG, J. 2006. Adaptive localized QoS-constrained data aggregation and processing in distributed sensor networks. *IEEE Trans. Para. Distrib. Syst. 17,* 9, 923–933.

Received March 2009; revised August 2009; accepted November 2009